REMARKS

Examiner

Claims 1-8 are rejected under 35 U.S.C. 102(e) as being anticipated by Nagarajan (U.S. Pat. No.6,522,791, hereinafter Nagarajan). Nagarajan teaches a digital scanner for scanning images and a method and apparatus for processing digital data accurately wherein all of the claimed circuitry is either explicitly or inherently taught as evidence by the fact that the digital scanner converts the original images into full-page image using a GDI establishing addresses corresponding to recording positions of the objects (please note: col 3, line 6 – col 5, line 23).

Response

Please amend the present application with the replacement paragraph specified above. In the first sentence of said paragraph, the identification number for the address units has been corrected from "74" to be "28". Additionally, the term "full page <u>image</u> 80" has been replaced with "full page <u>data</u> 80" throughout the paragraph. This change language change is supported at least by Paragraphs [0007] and [0018], and the title of the application as filed. No new material has been introduced.

The major difference between the scanner disclosed by Nagarajan and the present invention is the data format for a printer. The disclosed data format for a printer is commands and/or information delivered from the GDI, with the printer driver storing the input data into the array of address units first. The data can be inputted repeatedly due to a plurality of objects overlapping on application file cases. Finally, full-page data is made and the printer driver starts to render the input data into image raster data, transforming it into the printer full page data format for printing. The full-page data is the language, which is made for the printer driver to render out one full page of raster CMYK data into the printer data format for printer device to print.

Paragraphs [0022] and [0023], and Figs.4-6, define one embodiment of the present invention similarly to the following.

A document 70 comprises some objects 74 and the document 70 is divided into a plurality of preferably equal sized blocks 72. The memory 26 has a plurality of address units 28, each corresponding to one of the blocks 72 for recording positions of the objects 74 in the corresponding block 72. For example, the first address unit REC#0 records the positions of the objects 74 in the block 72 (perhaps the first ten scan lines of the document 70, i.e. the scan lines 0-9), and the second address unit REC#1 records the positions of the objects 74 in the next block 72. Initially all positions in all address units 28 are null (or another similar value) because no positions of any objects 74 have as of yet been ascertained.

The GDI 14 provides the objects 74 of the document 70 and the coordinates of the objects 74 in the document 70 to the printer driver 16, and then the positions recorded in the address units 28 are updated according to the coordinates of the objects 74 provided by the GDI 14. If the corresponding block 72 of the address unit 28 does not contain any part of the objects 74, a predetermined flag (i.e. a "block empty" flag) is written into the address unit. If the block 72 contains any part of the objects 74 of the document 70, the positions in that address unit 28 are updated to reflect the position of the objects 74 in the corresponding block 72.

For example, the first three blocks A-C (Fig.5) do not contain any object 74, so the corresponding three address units REC#0-REC#2 record the same flag, such as a null value, and the three corresponding images pieces #A-#C are generated by filling in a predetermined code (i.e. an "empty code"). On the other hand, the block N contains some objects 74, so the corresponding address unit REC#N records an upper left coordinate and a lower right coordinate of the block N in the document 70, and the image piece #N is generated by performing a conversion process on the block N according to the pixels of the objects 74 in the block N provided by the GDI 14.

Fig. 6 is a schematic diagram of the document 70 and the generated full-page image 80. The full-page image 80 has a plurality of image pieces 82, and each of the image pieces 82 corresponds to one of the blocks 72 of the document 70. The print data of the document 70 is generated according to the updated positions recorded in the corresponding address unit 28. For example, the image piece #A is generated according to the updated positions recorded in the address unit REC#0.

Because in this example, block A did not contain any part of the objects 74, the corresponding address unit 28 is marked as empty, the corresponding pixels in the generated image piece #A may all simply be set to null or a default background color. On the other hand, block N contained objects 74 and the corresponding address unit 28 now comprises the coordinates indicating the position of the objects 74 within that block. Therefore, the pixel values within those coordinates must be converted for the corresponding image piece #N. After all of the images pieces 82 of the full-page image 80 are generated, the images pieces 82 are merged into the full-page image 80, and then the full-page image 80 is transmitted to the printer 30.

In brief, the present invention allocates an array of address units, each corresponding to a predetermined portion, or block, of the document. The address units are updated to comprise positions provided by the GDI of objects within the corresponding block. Then an image piece corresponding to each address unit (and block) is generated according to the updated positions, or lack of updated positions, in the respective address unit. All image pieces are merged into a full-page image for printing. This summary is quite similar to the limitations found in claim 1 as filed.

On the other hand, referring to Fig.3 and the cited text, Nagarajan teaches an image processing unit 70 that coverts the analog image data of a scanned image into digital form. Full page video data is available from detection module 150 and such image data is defined in pixel form by segmentation module 150 (Col.4, lines 15-25). "An image data analyzer 160 is connected to receive the full page video histogram of the scanned image as well as the pixel tag histogram obtained from the segmentation module 152. The pixel tag histogram contains information with respect to the pixel

type, (i.e. one of the 32 classes of pixels)." (Col.4, lines 26-33).

Image data analyzer 160 includes a full page video image module 161 which operates to generate a histogram representing said video data. Further data based on the full page histogram is collected in video statistic module 162. As another part of the image data analyzer 160, a pixel tag histogram is generated in pixel tag module 163 and data based on the pixel tag histogram is collected in pixel tag statistic module 164. The processed image data from each source is combined and further processed in data analyzer 165 to determine optimum settings for both mode and specific parameters. The optimum settings are then compared to the current settings, in the scanning assist module 166, to generate recommended adjustments. The adjustment data may be stored in SRAM 140 and communicated to the user through the scanner interface 40 which may include workstation 50. By operating through the scanner interface, the user has the option to select the optimum settings (Col.4, lines 44-60).

With this in mind, the Applicant is unsure exactly what the Examiner is referring to when stating that Nagarajan teaches a device and method wherein all of the claimed circuitry is either explicitly or inherently taught as evidenced by the fact that the digital scanner converts the original images into full-page image using a GDI establishing addresses corresponding to recording positions of the objects. It is agreed that Nagarajan converts original scanned data into full-page images. However, the Applicant does not understand how the results and the method employed by Nagarajan utilized in this conversion meets all of the claimed limitations in the present application where GDI informational data is sent to the printer driver and all GDI data position information is recorded into array of address units, storing the GDI information data into memory until the end of the page when the printer driver starts rendering the GDI input data into raster image data which is transformed into the GDI printer data format and transferred to the printer to print.

For example, it is unclear where Nagarajan teaches dividing the document (or even scanned data of the document) into a plurality of blocks, unless the Examiner considers each individual pixel a separate block. Obviously, the word "block" as

utilized in this application does not mean individual pixels, but areas containing a plurality of adjacent pixels (Paragraph [0022], Figs.2,3, and 5-7). The use of blocks as described in the present application allows "the merging a plurality of image pieces, where each of the image pieces is generated without referencing other objects located within other blocks of the document. Moreover, at a given time, only one of the image pieces is generated so that the capacity of the memory for temporarily storing data can be reduced." (Paragraph [0025]). If the data area coming from the GDI can be recognized in the data block, it means the S/W module process can be focused on those blocks having data input to save data processing time. The applicant cannot find teachings of, or any reasoning why the division of a document into blocks is necessarily present in the cited prior art.

It is unclear where Nagarajan teaches using an array of address units, each corresponding to a specific block, each address unit comprising positions of objects within the corresponding block. Nagarajan does teach pixel tag histograms, however the "pixel tag histogram contains information with respect to the <u>pixel type</u>, (i.e. one of the 32 classes of pixels)." (Col.4, lines 26-33). The pixel types are utilized to generate optimal settings for presentation to the user. Such setting include "lighter/darker and contrast settings". Purely for the sake of discussion, it may possibly (but not absolutely necessarily) be assumed that some kind of method, such as an array of "pixel units", allowing access to individual pixels may be present in Nagarajan. Access to individual pixels may be used to change the contrast, for example, but there would be no reason for said "pixel units" to comprise updateable positions of objects within corresponding blocks, or of themselves if the pixels were to be somehow considered both "objects" and "blocks" which obviously is not the intended meaning of "blocks". The printer does not need to provide the optimized settings for a user. The final purpose for the printer driver is to provide better rendering speed for a user. Rendering speed is increased through the utilization of the array of address units to record all object data inputs in the information from the GDI. Please also note that the data is not pure image data, it is in Windows GDI data format, and the data format needs to be translated into the printer data format after full-page data generation by the GDI. The Applicant is unable to find any teachings or

suggestion of why it would be necessary for the prior art to include address units corresponding to blocks and comprising updateable positions of objects within that corresponding block.

It is unclear where Nagarajan teaches updating the positions recorded in the address units according to the coordinates provided by the GDI of the objects in the corresponding block. The Windows GDI will cause windows application files to deliver some information and data to the printer driver. That data will generate situations where the objects data overlap, causing the data to update more times in the array of address units. As previously discussed, Nagarajan appears to lack address units and blocks, so there is no possible reason to assume that positions would be recorded in the address units and are updated according to coordinates of the objects in the corresponding block.

It is also unclear where Nagarajan teaches the GDI delivering data to the printer driver, the printer driver recording and storing information including object positions into the address units. After one page of GDI data is recorded according to the GDI information, the printer driver starts to render the data based on final address units records to create the printer data format. The Applicant is unable to find blocks, updated positions, address units, image pieces for each block, or merging in the prior art, the disclosed method, nor any suggestion why they would be necessarily present in the Nagarajan.

In summation, the claimed invention allocates an array of address units, each corresponding to a predetermined portion, or block, of the document. The address units are updated to comprise positions of objects within the corresponding block and provided by the GDI. An image piece corresponding to each address unit is generated according to the positions, or lack of positions, in the respective address unit. All image pieces are merged into a full-page image for printing. It is an advantage of the present invention that each of the image pieces is generated without referencing other objects located within other blocks of the document so that the combination of the objects is efficient (Paragraph [0009]). Moreover, at a given time, only one of the

image pieces is generated so that the capacity of the memory for temporarily storing data can be reduced (Paragraph [0025]).

The Applicant has demonstrated many distinct differences between the claims of the current application and the cited prior art. The two inventive methods perform different functions, for different reasons, and have different results. There is no reason to assume that limitations of the present invention are necessarily present in the cited prior art. As such, the Applicant believes that the current application represents a new and useful disclosure and respectfully requests reconsideration and allowance of claims 1-8.

Sincerely yours,

Winston Hsu

Date: 8/20/2004

Winston Hsu, Patent Agent No.41,526
P.O. BOX 506
Merrifield, VA 22116
U.S.A.
e-mail: winstonhsu@naipo.com.tw

(Please contact me by e-mail if you need a telephone communication and I will return your call promptly.)